基于平滑 L_1 范数的深度稀疏自动编码器社区识别算法 *

张军祥，李书琴†，刘　斌

(西北农林科技大学 信息工程学院，陕西 杨凌 712100)

摘　要：大数据时代，利用传统的社区发现算法对大规模复杂网络进行社区结构挖掘显得愈发困难，准确率也较低。因此，提出一种基于平滑 L_1 范数的深度稀疏自编码器社区发现算法 L_1 -ECDA（community discovery algorithm for deep sparse self-encoder based on smooth L_1 norm）。该算法首先采用基于 s 跳的方法对网络图的邻接矩阵进行预处理；然后构建基于平滑 L_1 范数的深度稀疏自编码器，并通过训练网络图相似度矩阵得到低维特征矩阵；最后采用 K-means 算法对低维特征矩阵进行聚类得到网络社区结构。通过在仿真网络与真实网络数据集上实验表明，L_1 -ECDA 算法有效提高了社区识别的准确率，且比 DBCS 算法准确率平均高 4%，比 Deepwalk 算法和 CoDDA 算法平均高 5.4%。

关键词：深度学习；社区识别；稀疏自编码器；平滑 L_1 范数

中图分类号：TP391　　doi: 10.19734/j.issn.1001-3695.2018.09.0743

Sparse AutoEncoder community recognition algorithm based on smoothed L_1 norm

Zhang Junxiang, Li Shuqin†, Liu Bin

(*College of Information Engineering*, *Northwest A & F University*, *Yangling Shanxi* 712100, *China*)

Abstract: In the age of big data, it is increasingly difficult to make the community structure mining of large-scale complex networks by using the traditional community discovery algorithm and the accuracy rate is low. Therefore, this research come up with L_1 -ECDA, a community discovery algorithm for deep sparse self-encoder based on smooth L_1 norm. This algorithm preprocessed the adjacency matrix of the network diagram with the method based on s Jump; then it established the deep sparse self-encoder based on smooth L_1 norm and get the low dimensional characteristic matrix by training the similarity matrix of the network graph; Finally, it get the network community structure by clustering the low-dimensional feature matrix through the K-means algorithm. Experiments on simulated network and real network data set show that the algorithm of L_1 -ECDA improves the accuracy of community recognition effectively. Its accuracy rate is 4% higher than the DBCS algorithm on average, and 5.4% higher than Deepwalk algorithm and CoDDA algorithm on average.

Key words: deep learning; community recognition; sparse self-encoder; smoothing L_1 norm

0　引言

复杂网络由大规模用户个体及用户之间的复杂关系所构成，社区结构作为复杂网络的重要特征之一，往往社区内部节点之间的连接相对稠密，社区之间节点的连接相对比较稀疏[1,2]。现实世界中诸多网络都呈现出社区结构，比如高校学生由于兴趣差异而构成不同的社团关系网络、知网中学者之间通过论文引用形成关系网、电商网站中客户购买商品形成交易网等。近年来，社区发现研究引起了学术界相关学者的高度重视，在社会学、计算机科学等众多领域获得了极大关注与深入研究[3]。社区发现对复杂网络中节点内部关联、个性化推荐、舆情分析及信息传播具有重要研究意义。

近年来，整个互联网发展进入大数据时代，伴随着整个网络用户数量呈爆炸性增长，网络节点剧增，节点之间的关系越发复杂。比如，腾讯、阿里巴巴等用户规模早已超过 10 亿，Facebook 每月的活跃用户数量超过 13 亿。因此，对大规模复杂网络社区结构进行挖掘，分析用户之间的关联关系，发现用户的行为规律，可以为广告投放、精准营销、个性化推荐及舆论控制等提供辅助决策支持。

然而传统社区检测算法对于节点数量动辄上百万，节点之间关系错综复杂的大规模复杂网络进行社区结构挖掘准确率往往较低。因此提出一种更加准确、新型的大规模网络社区识别算法成了亟需解决的问题。从提高社区识别的准确率出发，本文提出基于平滑 L_1 范数的深度稀疏自动编码社区检测算法 L_1 -ECDA。该算法通过对网络高维相似度矩阵进行降维，将得到的低维特征矩阵进行聚类分析，从而得到更加准确的网络社区结构。 L_1 -ECDA 算法流程如图 1 所示。

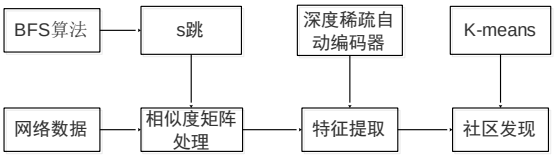


图 1　L_1 -ECDA 算法流程

Fig.1　L_1 -ECDA algorithm flow chart

本文主要贡献如下：

a）利用基于 s 跳数方法对网络节点的邻接矩阵进行预处理，处理后的矩阵既能反映网络拓扑结构中直接相连节点之间的相似性，又能反映不直接相连节点间的相似关系。

收稿日期：2018-09-23；**修回日期：**2018-11-19　**基金项目：**陕西省重点研发计划资助项目（2017GY-197）；中国博士后科学基金资助项目（2017M613216）；陕西省自然科学基金资助项目（2017JM6059）；陕西省博士后基金资助项目（2016BSHEDZZ121）

作者简介：张军祥（1992-），男，贵州铜仁人，硕士研究生，主要研究方向为智能信息系统；李书琴（1965-），女（通信作者），陕西渭南人，教授，博导，主要研究方向为智能信息系统(lsq7092417@126.com)；刘斌（1981-），男，陕西渭南人，副教授，硕导，主要研究方向为并行计算、计算机视觉、机器学习.

b）提出基于平滑 L_1 范数的深度稀疏自动编码器学习方法，提取相似度矩阵的特征表示，得到的低维特征矩阵对网络拓扑结构中的社区结构具有更好的表达能力。

c）通过在仿真数据集，Stanford 大学网络数据集及小规模数据集上实验表明，本文提出的 L_1 -ECDA 算法可以得到更加准确的网络社区结构。

1　相关工作

1.1　社区发现

设大规模网络图 $G=G(V,E)$，社区发现根据网络结构中的连接关系，将全部节点聚合成一系列子结构，即社区[4]。同一社区内节点之间的连接相对紧密，而不同社区之间的连接相对稀疏[1]。当前，经典的社区发现算法可以分为模块度优化法、标签传播法、图分割法与图嵌入法。

基于模块度优化算法主要思想是将社区发现问题转换为数学优化问题，通过将模块度定义为评价社区挖掘质量好坏的指标，对比模块度数值来得到最佳的社区划分结构。常见的算法有 GN 算法[5]、AGSO 算法[6]、Louvain[7]算法等。Louvain 算法将模块度优化与层次聚类相结合，使得算法的计算速度更快；同时，社区划分结果准确性也得到了提高。基于标签传播方法是一种启发式社区划分算法，其基本思想是根据已标记节点的标签信息去预测未标记节点的标签信息。典型的标签传播算法有 LPPB 算法[8]、MCPLP 算法[9]、COPRA 算法[10]、MMLP 算法[11]。MCPLP 算法首先计算未标记样本到标记样本间的最小代价路径，然后根据标记沿着节点间代价的最小路径传播来实现社区划分。He 等人[12]结合网络链路信息与标签信息，提出了一种基于多视图非负矩阵分解模型的社区发现算法，获得了较高质量的社区结构。基于图分割的社区发现算法是将图分割为两个子图，然后不断迭代，最后得出要求的子图数。Dilanni 等人[13]在考虑节点对之间互连量的情况下，引入 min-max 社区的概念，用于建模高度连接的节点集。Zeng 等人[14]在图分割理论基础上，研究了网络结构与并行聚类有效性之间的关系，提出了一种分布于内存机器上的并行社区发现算法。基于图嵌入方法是先对矩阵进行降维，再聚类得到社区结果。典型算法有 DeepWalk[15]、LE[16]、GraRep[17]。DeepWalk[15]算法根据随机漫步模型生成子网络，再利用 skip-gram 模型计算出网络图矩阵，通过聚类得到社区。

1.2　自动编码器

自动编码器(auto encoder, AE)[18]是神经网络的一种，其具有三层神经网络结构，即输入层、隐藏层及输出层。AE 通过将神经网络的隐藏层当做一个编码器与解码器，输入数据经过隐藏层后，到达输出层，利用反向传播算法来训练网络使得输入等于输出。其结构如图 2 所示，形象表示如图 3 所示。

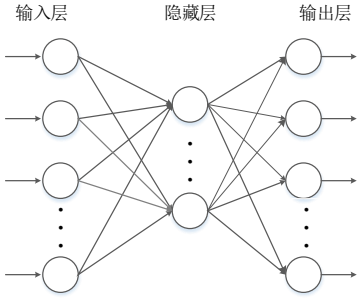


图 2　自动编码器结构

Fig. 2　Automatic encoder structure diagram

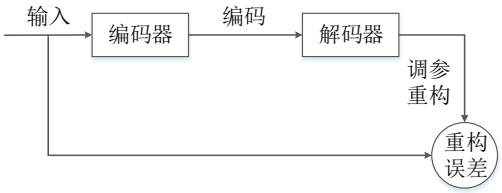


图 3　自动编码器形象表示图

Fig. 3　Automatic encoder image representation diagram

稀疏自动编码器（sparse Auto-Encoder,SAE）[18]是自编码器的一种,它是在自动编码器隐藏层神经元添加稀疏性限制条件而产生的一种衍生自编码器，能够在恶劣环境下学习到最好表达样本的特征，有效对数据样本进行降维。常见的数据降维方法有线性降维与非线性降维。线性降维在处理具有线性结构和高斯分布特征的高维数据时具有非常好的降维效果，但当数据集复杂且是非线性结构时，采用该类方法的效果往往不理想[19]。非线性降维方法在一定程度上存在短路边与领域参数的选择问题，同时在数据拓扑空间不稳定的情况下，容易受到噪声干扰。因此，需要借助其他技术来改进数据降维中存在的缺陷。本文采用深度学习中的稀疏自编码器进行数据降维处理，通过在自编码器基础上加入约束条件，使得自编码器每次迭代得到的表达数据尽量稀疏，从而通过少量的特征来表达输入数据，达到数据降维的效果。

2　数据预处理

对于网络图 $G=(V,E)$,其中 $V=\{v_1,v_2,\cdots,v_n\}$表示网络中节点的集合，$E=\{e_1,e_2,\cdots,e_m\}$为边的集合。节点之间的连接关系用邻接矩阵表示为 $Adj=[\omega_{ij}]_{n\times n}$，$\omega_{ij}$ 取值为 1 或 0，若 $\omega_{ij}=1$，则表示节点 v_i与v_j 之间存在连接关系，否则表示两节点之间不直接相连。若直接用邻接矩阵 Adj 来描述网络中节点之间的相似性关系，显然不全面，事实上网络图中不直接相连的节点也会存在一定的相似关系，仅仅使用邻接矩阵刻画网络图中节点的相似关系，显然会影响社区检测的质量。为了能够更加全面、真实地刻画网络图中节点之间的相似性关系，本文利用基于跳数的方法，对节点的邻接矩阵重新进行计算，得到节点新的邻接矩阵。

定义 1　跳数 s。设网络图 $G=(V,E)$，对于节点 $v,u\in V$，若节点 v 到节点 u 的最短路径为 s,则称节点 v 可以经过 s 跳到达节点 u。

定义 2　节点相似度。对于网络 $G=(V,E)$，其中 $v,u\in V$，则节点 v 与 u 之间的相似度 $Sim(v,u)$ 为

$$\mathrm{Sim}(v,u)=e^{\sigma(1-s)} \tag{1}$$

其中：$s\geq 1$,随着跳数 s 的增加，节点之间的相似度呈现先递增后减少趋势；σ 为衰减因子，$\sigma\in(0,1)$，它控制着节点相似度的衰减程度，σ 越大，则衰减越快。

定义 3　网络相似度矩阵。对于网络图 $G=(V,E)$，则定义图 G 对应的相似度矩阵为 $X=[x_{ij}]_{n\times n}$,其中 $x_{ij}=Sim(v_i,v_j)=e^{\sigma(1-s)}$。

数据预处理过程如算法 1 所示。

算法 1：计算跳数集合、网络图相似度矩阵

输入：复杂网络图 $G=(V,E)$ 的邻接矩阵 $Adj\in R^{n\times n}$,跳数阈值 s,衰减因子 σ。

输出：相似度矩阵 X。

1 for each　x　in　V；

2　初始化图 G 中所有的节点状态为未访问状态；

3　分别初始化跳数集合 $Hop=\mathrm{NULL}$；队列 $Queue=\mathrm{NULL}$；

4　将 x 设置为访问中状态，初始化 x 的跳数为 0，并将 x 和跳数 s，写入集合 Hop 中，并将 x 加入队列 $Queue$；

5 while　$Queue\neq\mathrm{NULL}$；

6　从队列 $Queue$ 中取出网络节点 u；

7　for each　v　in　$N(u)$；

8　　if　u 在 x 的(s-1)跳内且 v 处于未访问状态；

9　　将 v 设置为访问状态，同时将 x 到 v 的跳数等于 x 到 u 的跳数加 1；

10　　将 v 及 x 到 v 的跳数写入跳数集合 Hop，并将 v 加入队列 $Queue$；

11　　end

12 end

13 将 u 标记为访问结束状态；

14 end

15 for each　v　in　V；

16　根据跳数集合 Hop 及式(1)计算 x 和 v 的相似度 $Sim(x,v)$；

17 end

18end

19 Return 基于跳数的相似度矩阵 X；

在算法 1 中，先计算出跳数集合 Hop，再根据式(1)计算出网络图的相似度矩阵 X。从第 5 行到第 16 行，对网络图 G 中的每个节点 x，使用 BFS 广度优先遍历算法找到节点 x 在 s 跳内能达到的节点 v，将 v，x 与 v 之间的跳数写进集合 Hop，从 15 行到 17 行，计算 x 与点集 V 内其他节点的相似度，若 v 在 Hop 内，则使用式(1)计算 $Sim(x,v)$，否则 $Sim(x,v)=0$。

3　特征提取

本章将介绍 L_1 -ECDA 算法进行特征提取的详细过程。首先介绍稀疏惩罚函数平滑 L_1 范数；然后讲述构建基于平滑 L_1 范数的深度稀疏自动编码器的过程,对预处理后的相似度矩阵 X 进行特征提取，并通过聚类得到社区挖掘结果。

3.1　平滑 L_1 范数

为了能够更好地提取出高维数据中的低维特征值，往往使用稀疏惩罚函数对隐藏层的输出值加上某种稀疏性约束，从而实现为输入数据学习到稀疏表示。通常引入 KL 散度作为自动编码器中稀疏性的表示，其公式如式(2)所示。

$$S(t)=\rho\log\frac{\rho}{t}+(1-\rho)\log\frac{1-\rho}{1-\mathrm{t}} \tag{2}$$

其中：$\mathrm{t}=\frac{1}{m}\sum_{i=1}^{m}a_j^{(i)}$ 表示稀疏自编码器模型中第 j 个隐藏层单元在 m 个训练模型样本中的平均输出值；$a_j^{(i)}$ 为第 i 个样本的第 j 个隐藏层单元的输出值；超参数 $\rho>0$，表示稀疏级别，ρ 值越小则表示越稀疏。结合 KL 散度函数，得到稀疏编码器的目标函数如式(3)所示。

$$\min_{W,b}\frac{1}{m}\sum_{i=1}^{m}\frac{1}{2}h_{w,b}(x^i)-x^{(i)}{}_2^2+\beta\sum_{j=1}^{hs}S\left(\frac{1}{m}\sum_{j=1}^{m}a_j^{(i)}\right)+\frac{\lambda}{2}W_2^2 \tag{3}$$

其中：hs 表示稀疏编码器模型中隐藏层单元的个数。

虽然采用 KL 散度作为稀疏惩罚函数取得了不错的效果，但是根据稀疏理论，L_1 范数能够诱导出更好的稀疏性[20~22]，且已经广泛应用于机器学习与压缩感知领域[21]。但是并没有相关研究者使用 L_1 范数作为自编码器的稀疏性表示，来实现网络社区的检测研究。

学术研究过程中，之所以很少有研究学者使用 L_1 范数作为自编码器的稀疏项，其重要的因素是 L_1 范数在完整区间上是一个不可导函数，即在零点不可导，该缺陷给神经网络的优化带来了一定的挑战。针对此问题，本文对 L_1 范数使用“inf-conv”平滑技术来解决此问题，从而替换 L_1 范数作为自编码器模型的稀疏项。Abernethy 等人[23]提出一种称为“inf-conv”的平滑技术，L_1 范数作为不可微凸函数，能够很好地满足该平滑技术的条件。当“inf-conv”平滑技术的输入函数为 L_1 范数时，平滑 L_1 范数如式(4)所示。在式(4)中，L_1 范数与平滑 L_1 范数之间的相似度由超参数 μ 控制，若 $\mu>0$ 且取值越小，两者之间越相似。显然，引进平滑 L_1 范数实质就是在零点附近将不可导的 L_1 范数替换为 L_2 范数。

但是，若直接将 KL 散度式(3)更改为式(4)并作为稀疏自编码器的稀疏惩罚函数，则会出现一些问题，因为在稀

$$g_\mu(t)=\begin{cases}\dfrac{t^2}{2\mu} & |t|\le\mu\\ |t|-\dfrac{\mu}{2} & |t|>\mu\end{cases} \tag{4}$$

疏自动编码器中，常常选用 simoid 函数作为编码器的激活函数，其中 $\mathrm{simoid}(x)=1/1+\exp(-x)$，该函数在定义域内皆满足 $\mathrm{simoid}(x)\in(0,1)$。即此时 AE 中任意隐藏层的输出单元的输出值 $a_j^{(i)}$ 均满足 $a_j^{(i)}\in(0,1)$，进而导致惩罚函数的自变量 t 满足 $t\in(0,1)$。在该范围内，对 L_1 范数进行平滑是没有意义的，因为 L_1 范数函数在该定义域范围内是可导的。但由式(2)可知，引入稀疏级别参数 ρ 为稀疏惩罚函数带来了更大的可调性。基于此，本文将平滑 L_1 范数式(4)向右平滑 λ 个单位来克服上述“平滑无意义”的缺陷，且该做法可以在一定程度上提升平滑 L_1 范数在使用时的灵活度。因此，使用式(5)替换 KL 散度来作为稀疏自动编码器的稀疏惩罚函数：

$$S(t)=f(x)=\begin{cases}\dfrac{(t-\gamma)^2}{2\mu} & |t-\gamma|\le\mu\\ |t-\gamma|-\dfrac{\mu}{2} & |t-\gamma|>\mu\end{cases} \tag{5}$$

其中：为了保证平滑 L_1 范数的有效性，需要界定超参数 γ和μ 的取值范围，通常情况下，$0\le\gamma\le1$，因为 γ 是控制稀疏级别的，若 γ 在非定义域内取值，则针对 L_1 范数进行平滑操作是没有意义的；而参数 μ 的取值为 $0<\mu\le max\{\gamma,1-\gamma\}$，设置 $0<\mu$ 是由于 μ 控制着 L_1 范数与平滑 L_1 范数之间的相似度，但为了防止平滑 L_1 范数在社区检测时退化为 L_2 范数，则要求 $\mu\le max\{\gamma,1-\gamma\}$。

3.2　构建基于平滑 L_1 范数的深度稀疏自动编码器

在图 2 中，从输入层到隐藏层则对应于图 3 中的编码(encode)过程。当给定网络图 G 的相似度矩阵 $X=[x_{ij}]_{n\times n}$，输入其中一个节点在 G 中对应的向量 $x_i\in R^n$ 后，经过编码后，输出该节点对应的低维特征向量 $h_i\in R^d$。而从隐藏层到输出层则相当于一个解码的过程，在这个过程中，对低维特征向量 h_i 进行解码，得到输出向量 x_i'，且 x_i' 与 x_i 具有相同的维度。在编码与解码的过程中，使用反向传播算法训练网络，调整编码器与解码器中的参数，使得重构误差最小化，从而让输出向量 x_i' 与输入向量 x_i 近似相等。而在这个过程中，得到的低维向量 h_i 即作为特征结果。

在上述网络训练的过程中，假设将 x_i 输入到具有 d 个神经元的编码层中，经过式(6)后，得到低维向量 $h_i\in R^d$。

$$h_i = s_f(Wx_i + p) \tag{6}$$

其中：s_f 为激活函数，常取 $s_f = 1/1+\exp(-x)$；$W \in R^{d\times n}$ 为权重矩阵；$p \in R^{d\times 1}$ 为编码层中的偏置向量。

将向量 h_i 输入到解码层中，通过式(7)解码后，得到 $x_i^{'} \in R^{d\times 1}$ 作为输出结果：

$$x_i^{'} = s_g\left(\tilde{W}h_i + q\right) \tag{7}$$

其中：s_g 是解码器中的激活函数；$\tilde{W} = W^T \in R^{n\times d}$ 为权重矩阵；$q \in R^{n\times 1}$ 为解码层中的偏置向量。

在训练的过程中，通过调整自动编码器中权重矩阵与偏置向量四个参数 $\delta = \{W, \tilde{W}, q, p\}$，则最小化 $x_i^{'}$ 与 x_i 的重构误差为

$$\min_{W,\tilde{W},q,p} mize \sum_{i=1}^{n} s_g\left(\tilde{W}s_f(Wx_i + p) + q\right) - x_{i2}^2 \tag{8}$$

基于 3.1 节，现使用式(5)为自动编码器添加稀疏性限制，则构建基于平滑 L_1 范数的稀疏自动编码器的重构误差如式(9)所示。

$$L(\delta) = \sum_{i=1}^{n} s_g\left(\tilde{W}s_f(Wx_i + p) + q\right) - x_{i2}^2 + S(t) \tag{9}$$

构建基于平滑 L_1 范数的深度稀疏编码器由多层稀疏自动编码器组成，其结构如图 4 所示。

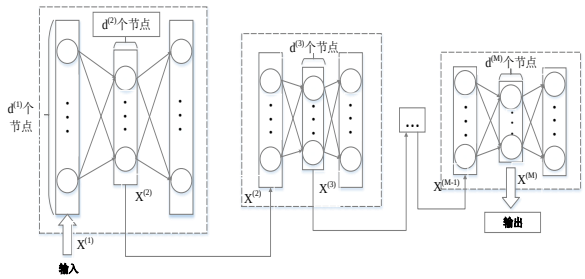


图 4　深度稀疏自动编码器结构

Fig. 4　Deep sparse autoencoder structure diagram

在训练的过程中，使用逐层贪婪的训练方法，具体的训练过程如下：首先设置编码器的层数 M 及每层的节点 $T = \{d^{(1)}, d^{(2)}, \cdots d^{(M)}\}$，其中 $d^{(1)} = n$，并将给定网络图 G 的相似度矩阵 $X^{(1)} \in R^{n\times n}$ 输入到具有 $d^{(2)}$ 个节点的编码器中，训练后得到编码结果为 $X^{(2)} \in R^{n\times d^{(2)}}$；然后将前一层得到的编码结果 $X^{(2)}$ 输入到具有 $d^{(3)}$ 个节点的编码器中，提取训练后的编码结果 $X^{(3)} \in R^{n\times d^{(3)}}$；如此循环，当最后一个自动编码器训练得到的编码结果为 $X^{(M)} \in R^{n\times d^{(M)}}$ 时，循环终止；最后，输出经过循环后得到的低维特征矩阵 $X^{(M)}$。特征提取详细过程见算法 2。

算法 2:对相似度矩阵 X 进行特征提取，再聚类得到社区结果。

输入：网络图相似度矩阵 X。

输出：社区发现结果 $\{C_1, C_2, \cdots, C_k\}$。

1 $X^{(1)} = X$；

2 for　j　=1 to　T；

3　构建基于平滑 L_1 范数的稀疏编码器；

4　输入特征矩阵 $X^{(j)}$；

5　通过优化式（4~8）训练稀疏自动编码器；

6　获得隐藏层的表示 $H^{(j)}$；

7　$X^{(j+1)} = H^{(j)}$；

8 end

9 对低维特征矩阵 $X^{(T)} \in R^{n\times d^{(T)}}$ 进行聚类，得到社区检测结果 $C = \{C_1, C_2, \cdots, C_k\}$；

10 Return　社区结果 $C = \{C_1, C_2, \cdots, C_k\}$。

在算法 2 中，从第 1 行到第 8 行是对相似度矩阵 X 进行特征提取。通过循环迭代 M 次，每次使用一个稀疏自动编码器从编码层中提取低维特征矩阵 $H^{(j)}$，并使 $X^{(j+1)} = H^{(j)}$ 作为下一次循环的输入矩阵。最终循环停止，得到低维特征矩阵 $X^{(M)} \in R^{n\times d^{(M)}}$。第 9 行使用 K-means 算法对低维特征矩阵进行聚类，首先以节点最小关联度原则选取新的聚类中心；然后以最大关联度原则进行模式归类，直到所有节点划分完为止；最后采用模块度优化确定 K 值，得到结果社区 $C = \{C_1, C_2, \cdots, C_k\}$。

4　实验结果与分析

本章中，首先对实验数据集进行简单的描述；然后介绍社区发现准确率的评判指标；最后针对社区发现的准确率、实验参数进行了详细分析，并对小规模数据集进行可视化展示。

4.1　数据集描述

本节采用三种数据集论证 L_1 -ECDA 算法的有效性：a)利用 LFR 基准程序随机生成人工模拟大规模复杂网络数据集[24]；b)Stanford 大学网络数据分析项目组 Stanford Network Analysis Project(SNAP) 真实复杂网络数据集(Http://snap.standford.edu/data/index.html)；c)利用典型的小规模真实网络数据集进行可视化展示。表 1 为实验室数据集详细信息，表 2 为 Epinionsl、NotreDame、Pokec 数据集的深度神经网络结构。

Lancichinetti 等人[24]提出 LFR 基准程序是一种用于生成模拟网络的算法。该算法可以用来验证社区检测算法的准确性，具有较高的实用价值。LFR 基准程序根据用户输入的参数，生成符合真实网络特征的人工合成网络与对应的社区结构。

表 1　实验数据集

Table 1　Experimental data set

a)仿真网络数据集

a) Simulation network data set

名称	节点	边	参数 γ	平均度
L-1W	10000	78546	0.3	15.71
L-10W	100000	2021456	0.3	40.43
L-50W	500000	9845687	0.3	39.38
L-100W	1000000	20254864	0.3	40.51

b)真实网络数据集

b) Real network data set

名称	节点	边	平均度	描述
Epinionsl	75879	508837	13.41	Epinions.com
NotreDame	325729	1497134	9.19	Notre Dame web
Pokec	1632803	30622564	37.51	Pokec 数据集
com-friendster	65608366	1806067135	55.06	Friendster-online social-network

c)小规模数据集

c) Small data set

名称	节点	边	平均度	描述
Karate	34	78	4.58	空手道俱乐部网络
football	115	652	11.33	足球队数据集
jazz	198	2742	27.00	爵士乐音乐家网络
facebook	5000	8194	3.28	Facebook 子网络

表 2　深度神经网络结构

Table 2 Deep neural network structure

数据集	每层节点数
Epinionsl	75879-61384-30692-16384-8192-4096-2048-1024
NotreDame	303516-151758-75879-61384-30692-16384-8192-4096-2048-1024
Pokec	1632803-1214064-607032-303516-151758-75879-61384-30692-16384-8192-4096-2048

4.2　评价指标及对比算法

本文用社区发现准确率 DA(dedection accuracy)与 NMI(normalized mutual information)这两个通用的社区评价标准对社区识别的准确率进行分析。社区发现准确率将查全率与查准率两个信息检索指标相结合，可信度较高。

定义 4　社区发现准确率 DA。将社区发现准确率定义为正确识别社区中节点的个数与网络节点总数的比率，用 DA(detection accuracy)表示，如式(10)所示。

$$\mathrm{DA}=\frac{\sum_{i=1}^{k} max\{C_i\cap C_j' | C_j'\in C'\}}{n},\ j=1,2,\cdots,k \tag{10}$$

其中：n 为网络节点数；$C=\{C_1,C_2,\cdots,C_k\}$ 表示原始的社区集合；$C'=\{C_1',C_2',\cdots,C_k'\}$ 表示利用算法检测出来的社区集合；$max\{C_i\cap C_j' | C_j'\in C'\}$ 为所有结果社区集与第 i 个精准社区 C_i 公共节点的数据的最大值。DA 值越大，则表示社区检测结果质量越好。

定义 5　NMI。归一化互信息(normalized mutual information,NMI),它是社区精准度评价标准之一，其计算公式如式（11）所示。

$$\mathrm{NMI}=\frac{-2\sum_{i=1}^{C_i}\sum_{j=1}^{C_j}N_{ij}\log\left(\frac{N_{ij}\times N}{N_{i.}\times N_{.j}}\right)}{\sum_{i=1}^{C_i}N_{i.}\log\left(\frac{N_{i.}}{N}\right)+\sum_{j=1}^{C_j}N_{.j}\log\left(\frac{N_{.j}}{N}\right)} \tag{11}$$

其中：C_j 表示第 j 个精准社区；C_i' 为第 i 个真实社区；矩阵 N 的行对应标准的社区结果，列对应算法得到的社区检测结果；$N_{i.}$ 表示第 i 行的求和，$N_{.j}$ 表示第 j 列的求和；N_{ij} 表示 C_j 与 C_i' 的公共节点数目。

实验对比算法包括尚敬文等人[25]提出基于深度稀疏自动编码器的社区发现算法 CoDDA，该算法与本文提出的 L_1 -ECDA 算法思路很相似，两者皆是先对相似度矩阵进行特征提取，再通过聚类得到社区结构。Deepwalk 算法[15]是一种基于图嵌入的方法，利用随机游走和 skip-gram 模型，获取到网络图的低维矩阵，再计算得到社区结果。DBCS 算法[26]是一种复杂网络社区发现并行算法，该算法采用模块度的思想，首先计算出节点之间的模块度增量，然后迭代寻找出所有模块度增量最大的节点对，再进行合并操作，并不断更新节点之间的模块度增量，从而实现大规模网络社区识别。

4.3　实验结果

4.3.1 社区发现准确率分析

由定义 4 可知，DA 能够直观表示社区检测的准确率，反映节点归属社区的正确性。因此，本文采用 DA 作为社区检测质量的评判标准。在表 3 和图 5 中列出了测试数据集在不同算法上社区检测的准确率。可以得出如下结论：

a）从整体看出，L_1 -ECDA 算法社区检测准确率相对于其他算法均较高，而且相对比较稳定，这是由于该算法在进行聚类前，采用了基于跳数的预处理方法，重新计算了网络节点的相似度矩阵，更加完善了节点的局部信息，并通过深度稀疏自编码器进行训练，得到更加准确表达社区结构的低维特征矩阵。L_1 -ECDA 算法划分的社区结果平均高出 CoDDA 算法 5.4%，这是由于在使用稀疏自动编码器进行特征提取时，L_1 -ECDA 算法采用平滑 L_1 范数作为自编码器的稀疏惩罚函数，得到的低维特征矩阵更能表达网络的结构，这证明了 L_1 -ECDA 算法的有效性。DBCS 算法在大规模数据集上准确率比 L_1 -ECDA 算法平均低 5.3%，但比 Deepwalk 算法平均高于 11.4%，且 DBCS 算法准确率能保持在 70%左右，识别效果较好，这也体现了并行计算的优势。Deepwalk 算法在状态转移过程中存在较强的随机性，且没有明确的优化目标函数，导致准确率整体较低，这也完全与文献[15]的情况相吻。

表 3　真实数据集上社区发现准确率对比/%

Table 3 Comparison of community detection accuracy on real data sets/%

	DBCS	Deepwalk	CoDDA	L_1 -ECDA
Karate	99.8	93.7	99.7	99.3
football	94.6	93.4	99.5	99.8
jazz	92.5	92.8	94.7	95.6
facebook	83.3	81.6	84.8	88.2
Epinionsl	75.5	64.8	78.7	83.3
NotreDame	70.6	59.7	68.6	74.7
Pokec	66.2	53.5	64.8	70.3
com-friendster	45.8	40.2	45.7	51.4

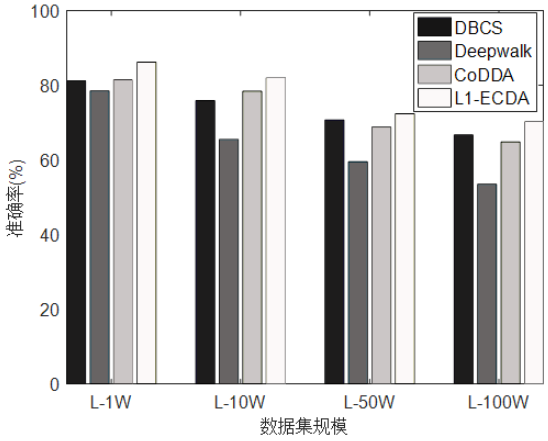


图 5　仿真数据集上社区发现准确率比较

Fig. 5　Comparison of community detection accuracy on simulation data sets

b）在图 5 中，当网络节点数目达到 10 万甚至更大时，整体准确率都逐渐呈下降趋势，这是因为随着数据集规模的扩大，网络中会出现越来越多的小社区，这会影响社区识别的准确率。对于 DBCS 算法，影响尤其重要。但在整体上，L_1 -ECDA 算法最终准确率能保持在 70%左右，这验证了本文提出算法的有效性。

生成仿真网络数据集的过程中，参数 γ 值控制着网络社区结构的明显程度，γ 值越大，则社区结构越不明显。在图 6 中采用 L-1W 数据集，随着 γ 值的不断增大，采用不同算法进行社区识别的准确率对比情况。

由图 6 可知，随着 γ 值的增大，DBCS、Deepwalk、CoDDA 及 L_1 -ECDA 算法的识别率都呈递减趋势，说明参数 γ 对社区发现质量具有较大的影响。当参数 γ 小于 0.3 时，四种算法的准确率相差不大；当 γ >0.4 时，L_1 -ECDA 算法的准确率明

显高于其他算法，这说明本文提出的 L_1-ECDA 算法，对于社区结构较为模糊的网络具有较好的性能优势。这是由于 L_1-ECDA 算法在特征提取过程中，取出有价值的信息，去除高维数据的冗余特征项，得到的低维特征矩阵更加能够表达节点的局部信息。

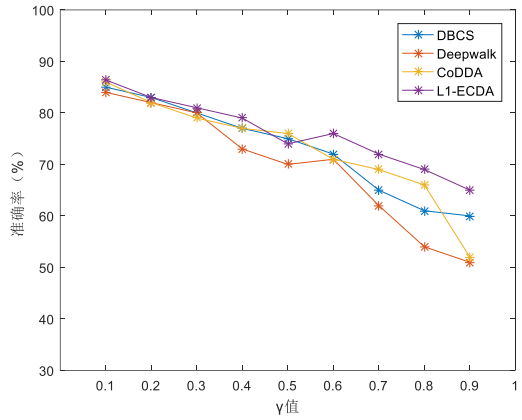


图 6　γ 值变化下不同算法社区发现准确率比较

Fig. 6　Comparison of community detection accuracy of different algorithms under change of the γ

4.3.2 参数实验

1）衰减因子参数 σ

对于 Epinionsl 网络数据集，设置跳数值为 s =15,构建基于平滑 l_1 范数的稀疏自动编码器每层的节点数为[75879-61384-30692-16384-8192-4096-2048-1024]，分析对比不同衰减因子参数 σ 的取值对于 NMI 的影响。由图 7 可知，使用 L_1-ECDA 算法对网络节点的相似度矩阵进行特征提取后，再进行社区划分比直接使用 K-means 算法进行聚类得到的社区划分结果更加准确。

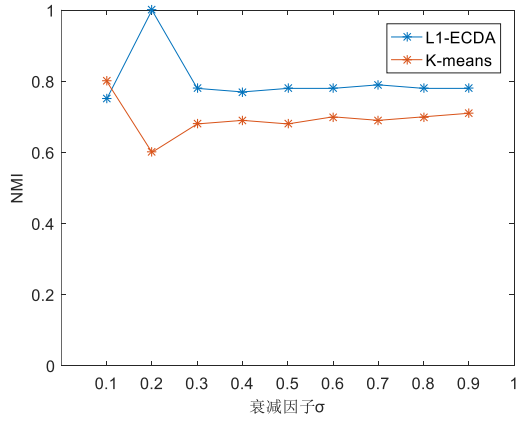


图 7　不同衰减因子参数 σ 下，在 Epinionsl 数据集上使用 L_1-ECDA 算法与 K-means 算法 NMI 值比较

Fig.7　Comparison of NMI between L_1-ECDA algorithm and K-means algorithm on Epinionsl data set under different value of attenuation factor parameter σ

根据图 7 所示，当衰减因子参数 σ 逐渐增加时，NMI 整体呈先递增再递减趋势。因为根据式（2）可知，随着跳数 s 的增加，节点相似度逐渐减少，而 σ 控制着相似度的衰减程度。对于规模较小的数据集 Karate，设置衰减因子参数 σ =0.6 来避免参数 σ 过大对社区边界的模糊作用。当数据集规模较大时，可以选择稍小的衰减因子 σ =0.2，这样可以更好的获取节点的局部特征，以达到最好的结果。

2）跳数参数 s

对于 jazz 网络数据集，设置衰减因子参数 σ =0.5,构建基于平滑 L_1 范数的深度稀疏自动编码器每一层的节点数为[198-128]，分析对比不同跳数的取值对于 NMI 的影响，并比较得出使用 L_1-ECDA 算法得到的社区划分结果比直接使用 K-means 算法进行聚类更加准确。

由图 8 所知，NMI 整体呈先递增后递减的趋势，这也符合实际情况，因为真实网络中，不直接相连但经过一定跳数可以达到的节点间存在一定相似度，若跳数过大，距离较远的节点也存在一定的相似度，却增加了社区识别边界的模糊度。对于规模较小的数据集 jazz，跳数阈值 s =3；对于规模稍微较大的数据集 facebook，选取跳数 s =9,即可以到达最优的结果。

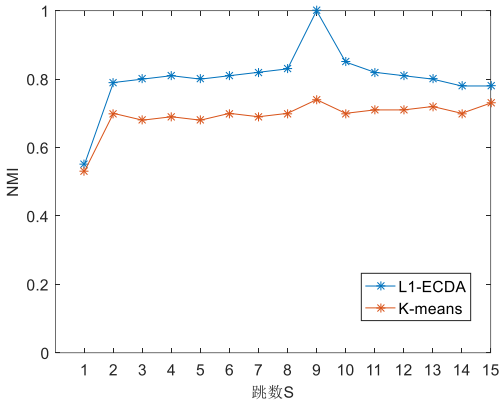


图 8　不同参数 s 下，在 facebook 数据集上使用 L_1-ECDA 算法与 K-means 算法 NMI 值比较

Fig. 8　Comparison of NMI between L_1-ECDA algorithm and K-means algorithm on facebook data set under different value of parameter s

3）编码器的层数 M

在构建基于平滑 L_1 范数的深度稀疏自动编码器基础上，对于数据集 Epinionsl、NotreDame，设置衰减因子参数 σ =0.1，跳数 s 分别为 15、20，对比分析不同层数的稀疏编码器对 NMI 评价指标的影响。

如图 9 所示，Epinionsl 与 NotreDame 数据集在不同层数的深度稀疏自编码器中采用 L_1-ECDA 算法的 NMI 值对比。对于 Epinionsl 数据集，当稀疏自编码器的层数达到八层（75879-61384-30692-16384-8192-4096-2048-1024）时，使用 L_1-ECDA 算法进行社区划分时性能达到最佳，但当深度稀疏编码器的层数再增加时，社区划分的准确性呈现递减趋势。结果表明，采用深度学习中的稀疏编码器学习方法可以提取网络社区结构中的特征信息，提高社区划分的准确性，但若编码器的层数设置过高，则可能部分特征信息被过滤掉，降低了社区划分的准确性。对于 NotreDame 数据集，当编码层数达到 10 层时，其社区划分质量达到最佳。

4.3.3 可视化展示

使用 L_1-ECDA 算法分别在数据集 Karate，football 以及 jazz 上进行实验并可视化展示。从图 10(a)~(c)可以发现，L_1-ECDA 算法在小规模数据集上的识别率很高，与经典的社区发现算法不相上下（表 3）。此外，根据图 10(b)与(c)可知，尽管 football 数据集与 jazz 数据集的节点数目相差不多，但 jazz 数据集的复杂度却比 football 高很多。由表 3 可知，采用 L_1-ECDA 算法进行社区检测时，football 数据集的识别率比 jazz 数据集要高 4.2%，这表明网络的复杂度对 L_1-ECDA 算法社区识别质量具有一定影响，显然与客观事实相符合。

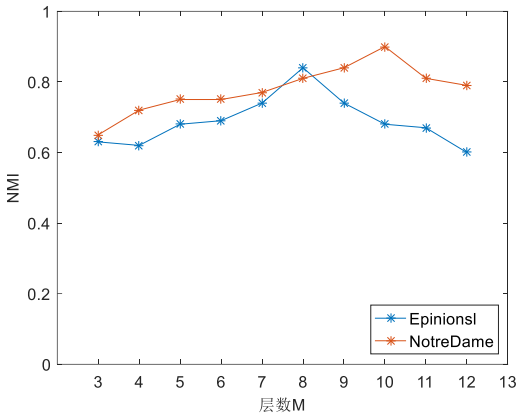
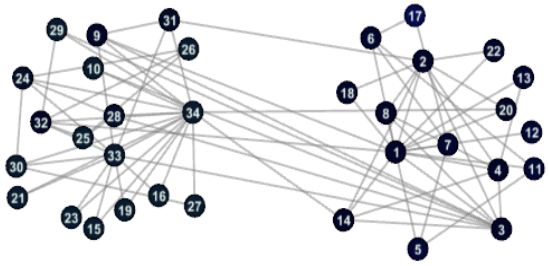


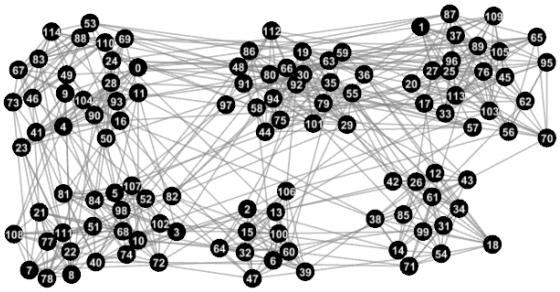
图 9　Epinionsl、NotreDame 数据集在不同层数的稀疏自编码器中使用 L_1 -ECDA 算法的 NMI 值

Fig. 9　Values of NMI from L_1 -ECDA algorithm on data sets of Epinionsl and notredame in sparse autoencoder with different layers



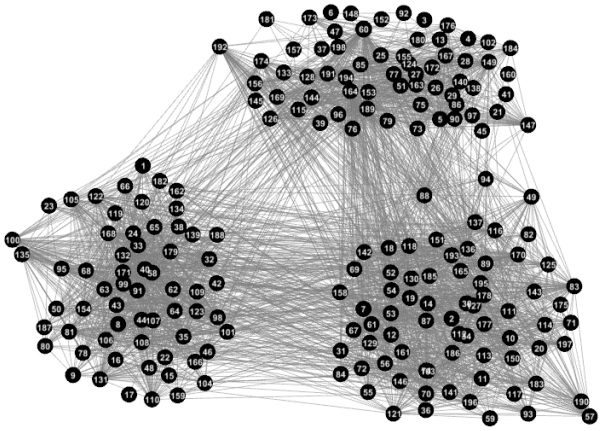
(a) karate 社区结构

(a) karate community structure



(b)football 社区结构

(b) football community structure



(c)jazz 社区结构

(c) jazz community structure

图 10　社区识别结果可视化

Fig. 10　Visualization of community recognition results

5　结束语

本文针对传统经典社区识别算法在大规模数据集上进行社区发现准确率较低的问题，提出了基于平滑 l_1 范数的深度稀疏自编码器社区发现 L_1 -ECDA 算法。该算法首先将网络图的邻接矩阵进行预处理，重新计算节点之间的相似度矩阵；然后使用基于平滑 L_1 范数的深度稀疏自编码器对相似度矩阵进行特征提取，得到网络图的低维特征；最后通过 K-means 聚类获得社区结构。通过在仿真数据集、Stanford 大学网络数据集及部分小规模数据集上论证，本文提出的 L_1 -ECDA 算法进行社区识别的准确性更高、稳定性更强。目前，网络重叠社区的识别能够更加真实反映网络结构特征，因此在后续的研究过程中，将重点研究复杂网络结构中的重叠社区结构。

参考文献：

[1] 李玉翔. 基于网络社区的用户兴趣建模与推荐技术研究 [D] 郑州: 信息工程大学, 2013. (Li Yuxiang. Research on user interest modeling and recommendation technology based on network community [D] Zhengzhou: Information Engineering University, 2013.)

[2] 付姣. 基于线图与标签传播的重叠社区发现算法研究 [D]. 武汉: 武汉科技大学, 2018. (Fu Wei. Research on overlapping community discovery algorithm based on line graph and label propagation [D]. Wuhan: Wuhan University of Science and Technology, 2018.)

[3] 周小平, 梁循, 张海燕. 基于 RC 模型的微博用户社区发现 [J]. 软件学报, 2014, 25 (12): 2808-2823. (Zhou Xiaoping, Liang Xun, Zhang Haiyan. Community discovery of weibo users based on RC model [J]. Journal of Software, 2014, 25 (12): 2808-2823.)

[4] Wang Meng, Wang Chaokun, Yu J X, *et al.* Community detection in social networks: an in-depth benchmarking study with aprocedure-oriented framework [J]. Proceedings of the VLDBEndowment, 2015, 8 (10): 998-1009.

[5] Newman M E J, Girvan M. Finding and evaluating community structure in networks [J]. Physical Review E, 2004, 69 (2): 026113.

[6] 冷作福. 基于贪婪优化技术的网络社区发现算法研究 [J]. 电子学报, 2014, 42 (4): 723-729. (Leng Zuofu. Research on network community discovery algorithm based on greedy optimization technology [J]. Chinese Journal of Electronics, 2014, 42 (4): 723-729.)

[7] 黄天诚. 基于图着色的并行 Louvain 社区发现算法研究 [D]. 长春: 吉林大学, 2016. (Huang Tiancheng. Research on parallel Louvain community discovery algorithm based on graph coloring [D]. Changchun: Jilin University, 2016.)

[8] 刘世超, 朱福喜, 甘琳. 基于标签传播概率的重叠社区发现算法 [J]. 计算机学报, 2016, 39 (4): 717-729. (Liu Shichao, Zhu Fuxi, Gan Lin. Overlapping community discovery algorithm based on label propagation probability [J]. Chinese Journal of Computers, 2016, 39 (4): 717-729.)

[9] 汪西莉, 蔺洪帅. 最小代价路径标签传播算法 [J]. 计算机学报, 2016, 39 (7): 1407-1418. (Wang Xili, Yan Hongshuai. The minimum cost path label propagation algorithm [J]. Chinese Journal of Computers, 2016, 39 (7): 1407-1418.)

[10] 孟令恒. 自动编码器相关理论研究与应用 [D]. 北京: 中国矿业大学, 2017. (Meng Lingheng. Research and application of automatic encoder related theory [D]. Beijing: China University of Mining and Technology, 2017.)

[11] Kim K H, Choi S. Label propagation through minimax paths for scalable semi-supervised learning [J]. Pattern Recognition Letters, 2014,

45: 17-25.

[12] He Chaobo, Fei Xiang, Li Hanchao, *et al.* A multi-view clustering method for community discovery integrating links and tags [C]//Proc of the 14th IEEE International Conference on e-Business Engineering. 2017. Melmaruvathur: IEEE Press, 2017:23-30.

[13] Di Ianni M, Gambosi G, Rossi G, *et al.* Min-max communities in graphs: complexity and computational properties [J]. Theoretical Computer Science, 2016, 613: 94-114.

[14] Zeng Jianping, Yu Hongfeng. A study of graph partitioning schemes for parallel graph community detection [J]. Parallel Computing, 2016, 58: 131-139.

[15] Perozzi B, Al-Rfou R, Skiena S. Deepwalk: online learning of social representations [C]//Proc of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. [S.l.]:ACM Press, 2014:701-710.

[16] Belkin M, Niyogi P. Laplacian eigenmaps for dimensionality reduction and data representation [J]. Neural Computation, 2003, 15 (6): 1373-1396.

[17] Cao S, Lu W, Xu Q. Grarep: Learning graph representations with global structural information [C]//Proc of the 24th ACM International on Conference on Information and Knowledge Management. [S.l.]:ACM Press, 2015: 891-900.

[18] Qiao Shaojie, Han Nan, Gao Yunjun, *et al.* A fast parallel community discovery model on complex networks through approximate optimization [J]. IEEE Trans on Knowledge and Data Engineering, 2018 , 30 (9): 1638-1651.

[19] 朱啸天，张艳珠，王凡迪．一种基于稀疏自编码网络的数据降维方法研究 [J]. 沈阳理工大学学报, 2016, 35 (5): 39-43. (Zhu Xiaotian, Zhang Yanzhu, Wang Fandi. Research on data dimensionality reduction method based on sparse self-coded network [J]. Journal of Shenyang Ligong University, 2016, 35 (5): 39-43.)

[20] Lee H, Battle A, Raina R, *et al.* Efficient sparse coding algorithms [C]//Advances in Neural Information Processing Systems. [S.l.]:MIT Press 2007: 801-808.

[21] 周巍．L1 范数最小化算法及应用 [D]. 广州：华南理工大学, 2013. (Zhou Wei. L1 norm minimization algorithm and its application [D]. Guangzhou: South China University of Technology, 2013.)

[22] 鲁亚平．面向深度网络的自编码器研究 [D]. 苏州：苏州大学, 2016. (Lu Yaping. Research on self-encoder for deep network [D]. Suzhou: Suzhou University, 2016.)

[23] Abernethy J, Lee C, Sinha A, *et al.* Online linear optimization via smoothing [C]//Proc of Conference on Learning Theory. [S.l.]:ACM Press, 2014:807-823.

[24] Lancichinetti A, Fortunato S. Limits of modularity maximization in community detection [J]. Physical Review E, 2011, 84 (6): 066122.

[25] 尚敬文，王朝坤，辛欣，等．基于深度稀疏自动编码器的社区发现算法 [J]. 软件学报，2017, 28 (3): 648-662. (Shang Jingwen, Wang Zhaokun, Xin Xin, *et al.* Community discovery algorithm based on deep sparse autoencoder [J]. Journal of Software, 2017, 28 (3): 648-662.)

[26] 乔少杰，郭俊，韩楠，等．大规模复杂网络社区并行发现算法 [J]. 计算机学报, 2015, 38: 1-14. (Qiao Shaojie, Guo Jun, Han Nan, *et al.* Parallel discovery algorithm for large-scale complex network communities [J]. Chinese Journal of Computers, 2015, 38: 1-14.)